

Recherche d'Informations - Compte rendu de projet

Introduction

Avant de rentrer dans le vif du sujet, nous allons expliquer en quelques lignes le but de ce projet. Il fournit la possibilité à travers un système de lignes de commandes de traiter une image au format jpeg afin d'en extraire des données telles que la texture ou les couleurs afin de les comparer avec les images présentes dans une base. Le résultat sera une liste d'image correspondantes aux critères de l'image passée en entrée.

Pour informations : les annexes sont disponibles dans le dossier **Annexes** joint à ce rapport.

Lancement des programmes

Avant toute chose, lancez la commande

```
$ make
```

Pour obtenir le résultat d'une requête par histogramme de couleur il faut lancer

```
$ ./create_histo.exe fichier_jpeg_requete fichier_html_sortie
```

Pour obtenir un résultat par clusters lancez

```
$ bash lancer_cluster_depuis_image.sh fichier_jpeg_requete
```

le résultat se trouvera dans cluster.html

Organisation Logicielle

Nous avons réalisé 3 programmes principaux. (le .exe est présent seulement pour les différencier)

- create_histo.exe
- create_histo_cluster.exe
- create_histo_combine.exe

Ceux-ci permettent la visualisation d'un fichier html en fonction des entrées fournies. Le plus abouti étant **create_histo_combine** car il permet de visualiser les différences de résultats entre une comparaison basée sur les couleurs, les formes, ou encore une combinaison des deux.

La majeure partie du programme est réalisée dans le fichier histogrammes.c. Celui-ci comprend la réalisation des histogrammes de couleurs et clusters, le calcul de score entre chaque élément d'une liste et un fichier requête, ainsi que l'ordonnement de ces listes de scores.

Afin d'obtenir une comparaison avec les fichiers de la base, il faut d'abord traiter le fichier requête. Pour cela, nous avons mis en place un script qui automatise les étapes à réaliser

pour obtenir un fichier au format requis pour l'analyse de formes et qui lance aussi la création du résultat avec cette image pour requete : **lancer_cluster_depuis_image.sh**. Le fichier obtenu sera présent dans le dossier **mes_sift** sous la forme *fichier.jpg.sift*.

Implémentations

Dans cette partie, nous allons présenter plus en détails certains points de nos programmes qui nous semble intéressant de par leur mise en place, ou simplement car ils sont un point clé de ceux-ci.

Création d'un histogramme

Pour créer un histogramme de **couleurs** nous avons choisi de faire un tableau de taille $NBIN^3$ et de l'initialiser avec des 0. Puis d'incrémenter selon la formule suivante :

$$hist[r * NBIN * NBIN + g * NBIN + b] ++;$$

avec r,g et b les valeurs récupérées dans la structure CIMAGE. Ensuite l'histogramme est normalisé en divisant chaque bin par la taille de l'image (image.nx*image.ny).

Pour créer un histogramme de **clusters** nous avons choisi de faire un tableau de taille NCLUSTER et de l'initialiser avec des 0. Puis de parcourir le fichier contenant les numéros de cluster en incrémentant la case correspondante (ie. valeurdecluster-1). Il est ensuite normalisé en divisant par le nombre de lignes lues.

Créations du binaire de tous les histogrammes

Cela est fait en passant en entrée un fichier contenant toutes les urls ou chemins des images de la base à traiter et en créant chaque histogramme et en l'écrivant avec fwrite en binaire dans un fichier de sortie dont le nom est donné en entrée.

Distance Euclidienne

Nous n'avons pas appliqué la racine, mais cela ne change rien à l'ordre final lors du classement des résultats (et c'est plus rapide à calculer).

Création des clés

En récupérant une image requete ou le .sift d'une image requete, le fichier binaire des histogrammes et un fichier listant les images de la base, on peut créer la structure KEY qui contient l'association du score (distance euclidienne des histogrammes) et du fichier correspondant.

Ensuite en appelant sort_keys, on applique simplement un quick sort sur la structure avec une fonction de comparaison propre à nos clés (instanciée dans les fichiers fournis)

```
#define NBIN 4
#define NCELL 64
#define NVAL 256
#define NCLUSTER 256

KEY* combine_keys(KEY* color, KEY* cluster, int nbdocs, float pct_color, float pct_sift);
```

Combinaison des couleurs et formes

Afin de pouvoir obtenir des résultats en mixant les scores obtenus selon les couleurs et les formes des images, nous avons mis en place une fonction `combine_keys` qui mixe deux listes de descripteurs en une seule suivant des taux définis dans la fonction. Cette fonction permet dans le cadre du projet de combiner les descripteurs de couleur et de formes, avec un choix concernant l'importance de l'un et de l'autre.

Résultats Obtenus

A partir d'une image de la base et rapport couleur/forme 20/80

Nous avons effectué des tests avec une image déjà présente dans la base et avons décidé de faire varier l'importance des différents paramètres. D'après l'image en annexe, nous pouvons remarquer que de manière générale, certaines images d'un des deux descripteurs sont conservées, mais également que d'autres avec certainement plus de cohérences communes sont présentes. Dans ce cas là, malgré le fait de conserver beaucoup les formes, et d'avoir un vélo en image requête, on retrouve peu de vélos en résultats, mais la couleur dominante bleue est souvent conservée.

A partir d'une image de la base et rapport couleur/forme 80/20

Sur la base d'une même image, nous inversons cette fois-ci le rapport couleur / forme pour obtenir l'annexe 2. Cette fois-ci on retrouve étonnement une bonne partie des meilleures images de la partie précédentes, mais celles qui suivent semblent beaucoup plus cohérentes avec les images présentes dans le cadre de la recherche par couleurs, sans énormément d'images qui ne font pas partie des résultats avec un seul critère. On peut en déduire que la couleur a un fort impact sur la recherche d'image en général, même si dans ce cas, nous avons peu d'images et que ce raisonnement sera sûrement peu fiable dans le cadre d'une collection contenant des millions d'images.

A partir d'une image bicolore extérieure et rapport couleurs/forme 50/50

Bizarrement les images résultantes obtenues ne sont pas aussi tranchées que l'image utilisée. En revanche, on retrouve neuf images sur dix contenant des vagues, ce qui est la forme exacte de l'image. Pourtant, quand on effectue la combinaison des deux images, on n'obtient aucune de ces images, ce qui paraît étrange étant donné la ressemblance entre les images de la base selon la forme et la requête.

Annexes

Annexe 1. Résultats Image 1 20/80

Annexe 2. Résultats Image 1 80/20

Annexe 3. Résultats Image 2 50/50